

UNITED STATES PATENT APPLICATION FOR:

METHOD AND APPARATUS FOR PROVIDING DYNAMIC GROUP
MANAGEMENT FOR DISTRIBUTED INTERACTIVE APPLICATIONS

INVENTORS:

GEORGE POPESCU

ZHEN LIU

SAMBIT SAHU

ATTORNEY DOCKET NUMBER: YOR 920030529US1

CERTIFICATION OF MAILING UNDER 37 C.F.R. 1.10

I hereby certify that this New Application and the documents referred to as enclosed therein are being deposited with the United States Postal Service on 11/26/03, in an envelope marked as "Express Mail United States Postal Service", Mailing Label No. EV287971571US, addressed to: Commissioner for Patents, Mail Stop PATENT APPLICATION, P.O. Box 1450, Alexandria, VA 22313-1450

Alberta Gamble
Signature

ALBERTA GAMBLE
Name

Nov 26, 03
Date of signature

MOSER, PATTERSON & SHERIDAN LLP
595 Shrewsbury Ave.
Shrewsbury, New Jersey 07702
(732) 530-9404

METHOD AND APPARATUS FOR PROVIDING DYNAMIC GROUP MANAGEMENT FOR DISTRIBUTED INTERACTIVE APPLICATIONS

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] Embodiments of the present invention generally relate to network communications. More particularly, embodiments of the present invention relate to communication management based on network and on application criteria.

Description of the Related Art

[0002] Data communications in large scale distributed interactive systems, such as virtual environments and online games, require efficient data path control. Typically, users have been grouped together according to their explicit interest subscription or on an interest function that is evaluated at the application level. Such groupings have been used to control communications over a multitude of possible data paths and to a multitude of users.

[0003] Because of the diverse communication requirements of applications and users, a number of different data communication architectures have been proposed. For example, publication-subscription architectures can provide efficient data filtering at the application level by having an application server group users (and their communication criteria) into communication-requirement based classes, and then selectively communicating using group-based data distributions. While publication-subscription architectures are beneficial if an application's users can be efficiently grouped, such architectures do not take into consideration or make allowances for optimizing data paths according to end-to-end performance requirements, which can be important in time and/or data loss sensitive distributed applications. Furthermore, if the number of publishers becomes very large, inherent network node limitations can create communication bottlenecks. Similar problems exist in other communication

architectures such as interest-based communication architectures in which users specify and enter interest-based cells that partition an application space. Such interest-based communication architectures have the additional drawback that the application space partition information must be distributed to all users, which creates problems when the application space is dynamically partitioned.

[0004] The publication-subscription architectures and the interest-based communication architectures are examples of distributed content architectures that are designed to support direct communications between a small number of applications and a large number of users. Such architectures have been based on a particular application having an application server or servers that directly controls data distributions between the application and its users. However, many solutions do not take into account network metrics for session management while optimizing communication network infrastructure according to real-time application constraints.

[0005] Therefore, a method of using network middleware to provide communication services for distributed interactive applications by using a hierarchical mapping of participating nodes in a multidimensional network/application attribute space and automatic interest-based session management would be beneficial.

SUMMARY OF THE INVENTION

[0006] An embodiment of the principles of the present invention generally provides for a method of data communications for collaborative interactive applications using both network attributes and user communication interest.

[0007] Embodiments of the present invention provide for methods of using network middleware to provide communication services for distributed interactive applications using a hierarchical mapping of participating nodes in a multidimensional network/application attribute space and automatic interest-based session management. Such embodiments can provide for automatic interest-based session management with session control performed by dynamic grouping of users according to their application-level communication interests.

Such embodiments can implement a session hierarchical structure that is replicated at network forwarding nodes and at higher levels in the hierarchy. Dynamic changes in membership can trigger modifications in the session hierarchical tree and at the forwarding nodes that participate in the dynamically changed session.

[0008] Embodiments of the present invention provide for multiple large-scale interactive sessions that are supported on one communication infrastructure by establishing a hierarchical index of session information that is replicated on a distributed mesh of overlay nodes. Application spaces are modeled as N-dimensional attribute spaces that are partitioned into smaller cells that are mapped to communication groups. Dynamic partitioning of application spaces into variable size cells, transparent migration of users between cells, and dynamic repartitioning of application spaces are supported by communication middleware.

[0009] Embodiments of the present invention implement a session and data distribution management method that uses a hierarchical structure and that indexes client's communication preferences and network attribute information. The indexing structure generates dynamic group membership lists that map into communication groups. Indexing is partially replicated at control nodes in a network overlay, thus allowing distributed management of group membership. Send/receive operations are decoupled through data distribution and the session indexing structure. Senders and receivers register their group communication interest to a parent node in the hierarchy, and then the session control structure aggregate client's interest and dynamically update the replicas at control nodes selected according to the changes in client interest. The indexing structure has self-managing properties for automatic clustering based on a client session, on data interest, and on dynamic partitioning of the session/data interest attribute space. Group membership is managed by querying the indexed session/data distribution management structure at each control node and mapping output to the communication groups.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] So that the manner in which the above recited features of the present invention can be understood in detail, a more particular description of the invention, briefly summarized above, may be had by reference to embodiments, some of which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of this invention and are therefore not to be considered limiting of its scope, for the invention may admit to other equally effective embodiments.

[0011] Figure 1 is high level block diagram of an embodiment of a network middleware device that is suitable for implementing dynamic group management in accord with the principles of the present invention;

[0012] Figure 2 illustrates network map information;

[0013] Figure 3 illustrates an application space and application space partitions;

[0014] Figure 4 illustrates a functional flow diagram of the group management method;

[0015] Figure 5 illustrates distributed partition indexing;

[0016] Figure 6 illustrates a dynamic group management method that uses multi-attribute clustering;

[0017] Figure 7 illustrates hierarchical group management using communication interest aggregation with proximity-based clustering of client nodes and interest aggregation in a hierarchical control topology; and

[0018] Figure 8 illustrates communication interest indexed in a hierarchical structure with membership lists that correspond to communication groups stored at the leaf level.

[0019] To facilitate understanding, identical reference numerals have been used, wherever possible, to designate identical elements that are common to the figures.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0020] The principles of the present invention relate to a hierarchical-based communication session management system that indexes client's communication preferences and network attribute information. Indexing is used to generate dynamic group membership lists that map into communication groups. Indexing is replicated at control nodes in the network overlay to allow distributed management of group membership. Send/receive operations are decoupled through data distribution and the indexing structure. Senders and receivers register their group communication interests to a parent node in the hierarchy. The session control structure aggregates client interest and dynamically updates replicas at control nodes which are selected according to the changes in registered client interest. The indexing structure has self-managing properties for automatic clustering based on client session and data interests, and dynamic partitioning of the session/data interest attribute space. Group membership is managed by querying the indexed session/data distribution management structure and mapping output to communication groups.

[0021] Figure 1 is a high level block diagram of an embodiment of a network middleware device 100 that is suitable for implementing dynamic session management in accord with the principles of the present invention. The network middleware device 100 comprises a processor 110 and a memory 120 for storing a dynamic session management program 130. The processor 110 cooperates with conventional support circuitry 132, such as power supplies, clock circuits, cache memory and the like, that assist in executing the dynamic session management program 130. The memory 120 further includes data structures 135. Those data structures are multi-attribute structures that are described subsequently. The memory 120 includes a computer readable medium for storing software for operating the middleware device 100.

[0022] It is contemplated that some of the method steps discussed below are implemented as software processes. However, as is well known, software

processes, and thus the method steps discussed below, may be implemented within hardware and/or in firmware. As such, the method steps described herein are intended to be broadly interpreted as being equivalently performed by software, hardware, or a combination thereof. The network middleware device 100 also contains input-output circuitry 140 that forms an interface between the various functional elements, application servers, and users that communicate with the network middleware device 100.

[0023] An embodiment of the present invention maps application and network level information to form a common abstraction where clients, network resources (network level objects), and application level information is represented as objects defined by multiple attributes. Network resources are comprised of virtual IDs and network attributes such as network position parameters, network fanout, and delays. Application objects include group membership objects that are comprised of a list of parameters for group identification, an index of network objects in the group; and virtual content objects that include IDs and content holders.

[0024] The maps containing application level information and network resources are used for controlling information dissemination according to application constraints. Examples of controlled information dissemination include partitioning a group object on server resources for distributed group membership control and managing group membership for overlapping sets of overlay nodes at control overlay nodes.

[0025] What follows makes reference to a dynamic session management of virtualized network object IDs such as servers s1 ... s4, clients c1 ... c8; the elements of the session management model are: the network 3D maps s1(s11, s12, s13), ... c1(C11, c12, c13),... containing the position of the servers and clients in the network space; the application/virtual space partitioning partitions P1 ... P16; the client partition mapping Client-Partition map describing the interest of the clients in application space partitions: c1(P1,P2); c2 (P6,P7,P8); c3(P1,P2,P5); the content objects: : o1 ... o10 ; the content objects partitioning/replication map: P1(o1, o2, o3); P2(o4), and the Server-Partition

map s1(P1, P2, P5), s2(P1, P2, P4). The maps are used to construct a multi-attribute indexing structure that indexes the network and application level objects; the indexed network and application objects are grouped such that to satisfy network/application constraints as response time (e.g. s1 satisfies the grouping constraint – the network delay to c1, c3 is bounded), capacity constraints. This result of the grouping process are membership lists containing network objects: e.g. (s1)(c1,c3), the partitions associated with each group: (P1, P2) and the application objects corresponding to these partitions (o1, o2, o3, o4).

[0026] Figures 2 and 3 illustrate how a session management and data distribution infrastructure uses network resource maps, application interest maps, and application object maps. Network resources are characterized by multiple criteria, including network delay attributes, forwarding capacity attributes, and object category (client, server, middleware, etc.). Figure 2 shows a “virtual” three-dimensional network delay space 200 having client nodes C1, C3, C4, C5, C6, and C8, and server nodes S1 and S2. While Figure 2 shows a three-dimensional network delay space, it should be understood that the actual number of dimensions is a function of the target average error of representing round trip time (RTT) distances by Euclidean network space distances. By using the “virtual” three-dimensional network delay space map, clients and servers can be serviced based on round-trip time delays to achieve a communication goal. For example, clients and application servers can be assigned to communication middleware nodes so as to minimize round trip time delays. Other network resource characteristics have similar virtual maps and can be used in similar ways to achieve specific communication goals.

[0027] While Figure 2 illustrates how network resources can be mapped to achieve specific communication goals, the principles of the present invention also manage communications based on the application space partitioning. For example, Figure 3 illustrates an application space 300 that is divided based on communication interests into 16 partitions. Partition P1 (element 302) and partition P2 (element 304) are of communication interest to a client C1, while partition P5 (element 306) is not of interest to client C1. Partitions P1, P2, and

P5 are mapped to application server S1. Furthermore, the communication interest represented by partition P1 include the content object o1. Thus, an application can be characterized by an application space 300 that can be partitioned into areas of interest to some clients, but not others, a given client may have interests in multiple partitions, and at least some of the partitions are made available to network overlay nodes.

[0028] Embodiments of the inventive communication session and data distribution management system make use of several types of virtual maps. A first type is content replication map in which the map indicates the partitioning of the content on available network resources. A cell in the content partition is a collection of content objects (e.g. files, data streams, media objects). A content replication map indicates the set of content cells that are replicated at each network resource (servers).

[0029] Another type of map is a communication interest (session) partition map. A session map indicates the partitioning of current communication interest subjects to show the communication interest subjects in each partition cell.

[0030] Yet a more general representation of is an application object map in which objects (communication interest objects, content objects, etc.) defined by an application are partitioned are partitioned in a space defined by a set of associated descriptors and are then mapped to network resources. The mapping of application objects partitioning into network resources can be done by allocating multiple partitions per network resource and replication of a partitions at multiple network resources.

[0031] The use of maps in a communication session and data distribution management system 400 is illustrated in Figure 4. As shown, communication interest partitions 402, content partitions 404, and virtualized network objects 406 are indexed to form a multi-attribute space 408. Based on that indexing group membership is managed 410. This is performed by clustering users/clients according to their communication interests and network objects attributes indexed the multi-attribute space 408. The result is a step 412 of forming a membership list that assigns users to middleware communication

nodes and an application index that assigns partitions to middleware communication nodes. Based on the membership lists and the application index, at step 414 the communications are managed using a hierarchical structure that is based on the application index and membership lists. This enables multicast communications by having application servers send specific communications to specific middleware communication nodes that have users that are interested in such communications, and by having those middleware communication nodes forward communications only to the users that are interested in such communications. The application servers know which middleware communication nodes to send what information to, while the middleware communication nodes know who is to receive that information.

[0032] Referring now to Figure 5, application space partitions can be mapped into a set of connected network nodes (control nodes) using a distributed indexing method 500 in which the mapping function assigns one or several partitions to a control node according to aggregated partition control load. The distributed index function includes the step 502 of partitioning the application space and the step 504 of constructing a partition-control topology. Then, at step 506 a map is formed in which partition domains are mapped to control nodes. An example control topology is the hypercube topology, which gives logarithmic efficiency (measured as number of routing path hops) for routing control messages between any two nodes. The size of the partition-control topology depends on the aggregated control load required to support the communication interest dynamics of the client nodes. The mapping (step 506) of application space partitioning to the partition-control topology is logical; the distance between two partition-control nodes routing is measured as the number of network hops (proportional to the control messaging overhead) rather than distance in network delay space. Each control node has low connectivity (logarithmic with the number of nodes in the control topology). Partition-control nodes forward the message using greedy forwarding. After mapping, at step 508 the partition information is distributed to the control nodes. The load of routing control messages is distributed uniformly among the control

nodes (when the control partition queries are uniformly distributed among partition-control nodes).

[0033] The mapping of communication interests into multicast groups uses a data structure designed for efficient query of multiple parameters. The output of such queries is a membership list of network objects (identified by their virtualized network ID), which are mapped into communication groups. The mapping can be static, in which case static IDs are allocated for each session, or dynamic, in which case communication groups are formed at run-time according to a dynamic interest based grouping of network objects.

[0034] For applications with static group memberships, an efficient method of mapping interest groups into multicast groups is by allocating unique group IDs and by indexing the resulting distribution tree that corresponds to each group ID. This method clusters group members based on multi-attribute clustering and assigns a group ID for each cluster. While this multicast method makes use of the fact that the number of clusters is fixed, changes in node communication interest requires iterations from the current clustering solution, and changes in group membership triggers modification of the multicast distribution trees and modifications of the corresponding multicast group indexing structure. Similarly, changes in the network attributes of overlay nodes results in reconfiguration of the distribution tree and multicast indexing structure.

[0035] When the group membership is static, the indexing structure is an extensible table that contains the root of the distribution tree that corresponds to the group ID table entry. The indexing structure supports retrieval of the distribution tree by using the group IDs. Changes of the tree corresponding to a specified group ID can include deletion of a group ID, insertion of a group ID, and/or creation of a new distribution tree from a list of nodes.

[0036] In contrast to static mapping, dynamic mapping of interest groups that are identified by a set of descriptors into multicast groups is achieved by including with the routed data a set of group descriptors that are used to query the communication interest indexing structure to retrieve the set of group members that correspond to the described communication interest. The

multicast group membership list is constructed by the session management structure and passed to a communication engine, which construct the distribution tree topology. The distribution tree corresponding to a multicast group is comprised of multiple segments, each constructed by a control node that has local multicast group members. The group descriptors and the corresponding group members may be cached at control nodes in the data path such that subsequent data packets with the same set of group descriptors do not require reconstruction of the distribution tree topology. The mapping of interest groups into multicast groups can have an expiry policy in which the mapping between a set of parameters and a distribution tree is deleted when a cached entry is not used for a period of time.

[0037] Dynamic session management using multi-attribute clustering can be achieved by a session controller that uses mapping information to generate dynamical group membership lists. Grouping nodes according to their communication interest can be performed to improve or optimize constraints such as forward capacity, round trip time delay, communication efficiency constraints imposed by the application and network infrastructure, and data loss. The application constraints can be modeled using the metrics of delay penalty (delay in excess of the end-to-end delay maximum delay between overlay nodes in the same group), node maximum fanout (the maximum forwarding capacity of each node), grouping waste (the amount of wasted communication capacity due to sub-optimal grouping), batching window (the size of the batching window used for aggregating client interest), a look-ahead window (the size of the look-ahead window used for client interest prediction), and a look-ahead factor (a weighting factor used in the look-ahead window).

[0038] Inputs to an optimal clustering algorithm include defining interest membership vectors (a list of users/receivers that are interested in each application partition) and an objective (energy) function (a weighted sum of delay penalty, fanout utilization and grouping waste over all clients). Metrics that are useful for measuring the overall efficiency of the membership grouping method include the total fanout utilization (the sum of resource node utilizations for the current grouping considering each group as a unit of load) and the total

grouping waste (the sum of communication waste over all nodes in the communication group).

[0039] Based on the foregoing, a dynamic group management method that uses multi-attribute clustering is illustrated in Figure 6. That method includes a step 602 of registering node communication interests. A step 604 uses the registered node communication interests to form a hierarchical indexing of communication interests. The method 600 further includes a step 606 of registering node network information, followed by a step 608 to form a hierarchical indexing of the registered node network information. Based on the indexes formed in steps 604 and 608, the method 600 proceeds at step 610 with clustering nodes using dynamic group management algorithms. At step 612 cell membership lists are generated that correspond to each communication interest group. Finally, at step 614 a multicast group topology is formed.

[0040] Due to the mobility in a virtual application space, distributed interactive applications are characterized by high dynamics of client/participant communication interest. A session in a distributed interactive application is associated with each instance of an application. Sessions are decomposed into partitions; each partition grouping involves several cells (units of the virtual application space controlling the group membership of the clients). The dynamics of client interest requires a session management solution that controls client group membership with a minimum amount of signaling. Since the same communication infrastructure simultaneously supports multiple sessions, an effective session management solution needs to scale with the number of sessions and with the number of participants, particularly when each session may involve thousands of participants. The interactive nature of a distributed application imposes constraints on the grouping of participants and on the mapping of the participants into multicast groups that attempt to minimize the end-to-end delay between session participants or to minimize the end-to-end loss rate.

[0041] To provide scalable communication management the clients in a virtual space are clustered according to their common interest and according their network attributes. To do so, modeling of the session membership management is useful. Clients accessing services offered by the communication overlay specify their interest by defining a multi-dimensional domain in the application attribute space. In addition, clients are assigned a unique identifier associated with a set of network measurement attributes. Each client then can be mapped into a multi-dimensional space that spans network and session interest coordinates.

[0042] The attribute space contains network attributes, including network distance maps and network capacity maps, and application semantic attributes of client's communication interest (e.g. a multidimensional domain in virtual space attributes).

[0043] According to the principles of the present invention several methods of session membership control are available. For example, dynamic session membership that uses communication interest descriptors to express a client's interest, and explicit session membership control in which clients change their session interest profile by signaling a parent control node and receiving identifiers for corresponding session/sessions.

[0044] A session control method uses either a fixed partitioning of the application space, which is computed centrally at a session agent and replicated at control nodes, or a dynamic partitioning which computes session membership based on client interest and network metric constraints.

[0045] In addition to hierarchical indexing of communication groups, group management can use hierarchical aggregation of communication interest. For example the communication interest can be aggregated hierarchically on the overlay control topology. The hierarchical indexing describes the logical relation between overlay nodes. At each control node, the hierarchical structure indexes only a subset of the communication objects (overlay nodes) – the children and siblings of the control node.

[0046] As an example, Figure 7 illustrates hierarchical group management using communication interest aggregation with proximity-based clustering of client nodes and interest aggregation in the hierarchical control topology. Control node s1 indexes the communication interest of clients {c1,c2,c3,c4} and propagates control messages containing descriptors for active groups in the control hierarchy - e.g., when client node c1 sends a join message indicating its communication interest, s1 inserts a new group with descriptor SD1, adds client c1, and propagates the control message to s4 in the control hierarchy. The control node s1 indexes the membership lists: ML1(SD1)={c1} and membership list ML2(SD2)={c2,c3,c4,s2,s3}; The control node s3 indexes the lists: ML(SD3)={c6, c7} and ML4(sd2)={C5,S1}. The control message to s4 cause s4 to add s1 to the group for SD1 (inserting a new group for SD1 in case there were no previous members with such interest) and propagates control messages further in the hierarchy, thus aggregating the communication interest of the lower level nodes. S1 also maps the membership lists corresponding to group descriptors SD1 and SD2 into multicast groups. In Figure 8 the communication interest is indexed in a hierarchical structure with the membership lists corresponding to communication groups stored at the leaf level. Membership lists indexed at different nodes may refer to the same descriptor; since the data control uses the session descriptors, the membership lists can be local to control nodes. The communication interest index is queried at control nodes to retrieve membership lists composition; e.g. the query on SD2 at node s4 retrieves {s1,s2,s3}, while a query in SD2 at s1 retrieves client list {c2,c3,c4} and forwarding server list {s2,s3}.

[0047] The topological relation between nodes with the same group communication interest is constructed and modified dynamically by the communication infrastructure (communication engine). The leaf level nodes in the hierarchy are computed by clustering based on user preferences (e.g. transmission rate), network constraints and network topology rank. For each group of nodes with the same application level interest, the communication graph is constructed in a distributed manner by nodes in the control hierarchy, each node controlling the part of the communication graph containing child and

sibling nodes only; and each control node optimizes the construction of the communication path independently by using the network and user preference attributes returned by the interest indexing control structure. In addition to modifications imposed by group membership changes, the communication path can be modified without controlling the state of group membership at other overlay nodes.

[0048] The independence of session control from data path construction can be obtained using a combination of state-based or stateless mechanisms. A state based control requires signaling between control nodes, and is therefore used when data path changes often. The stateless control, used for high group membership dynamics, reduces the signaling overhead at the expense of per message processing overhead at each forwarding node. The scalability is increased since communication path can include stateless forwarding nodes. Dynamic change of data-path requires little overhead when processing of data path construction algorithms at the control nodes. In a network overlay with hierarchical aggregation of communication interest, the stateless control is advantageous at the leaf level (or levels at the bottom of the hierarchy) where the dynamics of the communication interest is high, while top level of the hierarchy, where dynamics is reduced due to interest aggregation, could use a state based group membership control.

[0049] The control of group management uses network space and logical (communication interest space) partitioning. Groups are managed based on a dynamic grouping of participants. Nodes at the higher levels in the hierarchy aggregate sub-domains of the virtual maps. Dynamic management of the groups comprises:

[0050] dynamic group management based on proximity on the virtual maps;

[0051] dynamic mapping of interest groups into multicast group; and

[0052] distributed control of group information based on multi-type attribute indexing of communication interest and network resource parameters.

[0053] The communication interest space is partitioned and mapped on a logical control topology. Each partition-control node indexes all control nodes with clients whose communication interest overlap with the partition. Each control node (aggregator of client communication interest) connects to a partition-control node, which route the control messages according to the communication interest descriptor.

[0054] The client nodes (senders and/or receivers) attach to the leaf nodes in the network control hierarchy. The control nodes maintain a hierarchy of the partition domains describing the virtual map. The map is partitioned into sub-domains which can be recursively split depending on the traffic between leaf nodes. The communication group indexed at a control node contains all client nodes proximal (in the network space) to the control node, participating in the same group, and the sibling nodes in the data dissemination/forwarding layer with clients participating in the group. The group membership is therefore controlled at several nodes in the forwarding/control layer. The membership lists may overlap: nodes in the forwarding/control may appear in multiple membership lists.

[0055] Using local group descriptors instead of global group IDs allows controlling of the scope of group identifiers (group are managed locally by control nodes). The control messages contain a list of descriptors of the inserted/deleted groups and require query insertion of control information indexing structure at each visited node.

[0056] The data dissemination also uses descriptors instead of global identifiers to identify at each control node the list of nodes participating in the same interest group. The virtual ID of these members is obtained by traversing the hierarchical group structure mapped at the leaf node. The leaf node then performs data forwarding to the nodes that participate in the same group, which is forwarded further to receiver nodes.

[0057] The group management method described here is based on replication and aggregation of group information at control nodes organized in a hierarchical topology. The participating overlay nodes are identified by virtual

IDs assigned according to their network attributes. The dynamic group management is designed symmetrical with respect to receiving and sending operations. The hierarchical aggregation of interest reduces exponentially the signaling between control nodes. Dynamic group management involves the following operations:

[0058] Group creation;

Group deletion;

Node joining; and

Node leaving.

[0059] Group creation/deletion:

[0060] Local groups are created/deleted based on the output of the local client node clustering. Groups are also deleted when the membership lists become empty; new local groups are created when a client node specify in the join request a communication interest for which there is no membership list indexed at the control node. Local group creation is followed by a distributed group look-up to find the control nodes with communication groups for the same communication interest and announce at the same time the identity of the new control node that created the local group. Local group deletion triggers a distributed group remove in the control hierarchy to announce the interested nodes the leave of the control node that deleted the group.

[0061] Receiver/sender node join:

[0062] The node join/leave messages are sent to the control node handling the client in the network topology. The “node join” triggers at the control node the insertion of the virtual node identifier in the membership list corresponding to the communication interest described in the join control message. If a group corresponding to the specified communication interest does not exist, a new local group is created as described above.

[0063] Upon receiving a node leave message, a control node delete the virtual ID of the sender from the group membership list corresponding to the specified

communication interest. As noted above, this triggers a group deletion when the leaving node is the only member of the group.

[0064] Algorithms for node join/leave and group creation/deletion.

[0065] The SD is the descriptor list which describes the communication interest of the joining/leaving node. The joining/leaving node has the virtual ID cID; the parent/control node of client cID has the virtual ID pID.

[0066] Node Join

Client cID sends a join [SD] message to the parent pID

At the parent:

Query indexing structure to retrieve the group entry corresponding to (SD);

If (no entry for SD)

Create_group(pID, SD);

Insert cID in the membership list of group (SD);

else insert (cID) in the membership list of group (SD).

[0067] Node Leave:

Client cID sends a remove [SD] control message to the parent pID;

At the parent:

Remove the ID of the leaving node entry from the group with descriptor SD;

If (Membership List of group (SD) == NULL);

remove group with descriptor SD

Delete_group (pID,SD).

[0068]

[0069] Create_group (nodeID, SD):

1. Insert a group corresponding to (SD) in the group indexing structure;

I. Control message propagation:

Propagate up:

a. Send control messages including the node ID and the

group descriptor to the parent node; stop if root of the control hierarchy;

b. Insert the node ID received at the parent node in the membership list corresponding to group (SD);

c. If SD membership list contains other members besides the one just inserted goto e.

d. Update the propagated ID with that of the current node and update the group descriptor and repeat from a.

[0070]

e. Propagate down:

f. Propagate the control message including the node ID and the group descriptor SD to all nodes listed in the SD entry except for the originator;

g. if (current node is leaf): insert the ID of the originator node in the SD entry; reply to the originator with the ID of the current node; stop.

h. else update the propagated ID with the ID of the current node and repeat from e.

[0071]

II. Tree processing : At originator node:

Insert the ID in the membership list corresponding to SD entry when receiving a reply from node ID.

[0072] Delete_group (nID, SD):

I. Message propagation:

1. Query the group indexing structure to retrieve the membership list corresponding to SD;

if (membership list is not null):

Propagate leave message with the node ID and the group descriptor to:

nodes in the SD membership list; delete the nID;

parent of the current node, recursively until reaching the root of the control hierarchy, updating the forwarded node ID at each step;

II. Grouping index processing:

At each node receiving the delete_group message: remove the node ID of the node who propagated the delete message from the SD;

[0073] While the foregoing is directed to embodiments of the present invention, other and further embodiments of the invention may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.